Diabetes Prediction using Machine Learning Techniques

Mark Dakroub[1], **Harishankar Murugan**[2], **Melvin Ravi**[3], **Nandhakumar Ragupathy**[4], **Peyanan Traore**[5] **and Justin Djidonou**[6]

[1]*Machine Learning*

Professor Christophe Bécavin, PhD, Université Côte d'Azur

Abstract—Diabetes is an international illness which is caused by a high glucose level in the human circulatory system. Diabetes should not be ignored nor should it be taken as a joke. If it is not treated it may cause some major issues in the human body. Being untreated, it can cause heart related problems, kidney damage, blood pressure, eye damage and affects vital organs such as the liver. Diabetes can be managed if detected earlier and treated with the right life style. The goal of this study is to create a machine learning model with such high accuracy that can predict the likelihood of someone having diabetes in the future. This study evaluates the performance of various machine learning models in predicting diabetic patients and non-diabetic patients. The machine learning models used are, Linear Regression, Logistic Regression, Decision Tree, Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), and Gradient Boosting, were all tested using a dataset TAIPEI diabetes dataset. After several experiments, the accuracy of each model turned out to be different compared to other models. The final result of this study shows that such a model with great accuracy exists and can be used for early diabetes prediction, the model is the Gradient Boosting Classifier, which achieved the highest accuracy of 95.24% compared to other machine learning models.

Github: https://github.com/Peyanan/Diabetes$_prediction.git$

Application Demo:$http\ :\ //ec2-13-53-216-86.eu-north-1.compute.amazonaws.com:5000/$

Keywords—*Diabetes, Machine, Learning, Prediction, Dataset, Ensemble*

1. Introduction

Diabetes is a rapidly growing and major cause of death among young people and the elderly. According to 2017 statistics, 425 million people are diagnosed with diabetes, 2-5 million patients lose their lives every year, and this number is increasing to 629 million [9]. Therefore, early predictions are important for taking early precautions. To accomplish this task, a dataset containing diabetic and non-diabetic patients is used to explore several intelligent models to train algorithms to help us predict prediabetic patients. We need to understand what happens in the body without diabetes.

Glucose comes from carbohydrates (pasta, rice, bread, cereal, fruit, dairy products, and starchy vegetables). Glucose moves in the bloodstream around our body to provide us with the main source of energy. Some are taken to our brain for cognitive functions, and the remainder is taken to our cells and stored in the liver for later use. Insulin (like a key to a cell door) is a hormone produced by beta cells in the pancreas that helps us transfer this glucose to our cells. Health problems arise when the pancreas cannot produce enough insulin (insulin deficiency), useless insulin (insulin resistance), build-up in the bloodstream (hyperglycemia) or diabetes mellitus.

Type-1 diabetes mellitus (DM1) is an autoimmune disease that destroys pancreatic beta cells, leading to insulin mutation or inadequate insulin secretion, causing insulin deficiency and hyperglycemia.

Type-2 non-insulin-dependent diabetes mellitus (NIDDM), it is where the body will not produce insulin properly and in sufficient amounts, causing the body to be resistant to insulin.

Type-3 gestational diabetes, where women develop high blood sugar with a high probability of type 1 and type 2 diabetes.

The objective of this study is to build an early prediction model for the diagnosis of diabetes. We will compare different models (Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbor, Gradient Boosting) to get the best accurate prediction. These experiments will leverage the use of machine learning algorithms for future early predictions which can help boost the physician's confidence for diagnosis.

2. Literature Review

The analysis and predictions were performed using various methods and techniques. Dr. Saravana Kumar N M, Eswari, Sampath P, and Lavanya S (2015) used Hadoop and Map Reduce technique which predicts diabetes and risks associated with it. This system is economical for any healthcare organization. [5] Yasodhaet al. [3] uses classification on various types of datasets gathered from a warehouse containing 200 instances of blood and urine tests with nine attributes. These instances of this dataset. K. Rajesh and V. Sangeetha (2012) used classification technique called C4.5 decision tree algorithm to increase classification efficiency. [2] Humar Kahramanli and Novruz Allahverdi (2008) used an ANN) with fuzzy logic to predict diabetes. [6] Mani Butwall and Shraddha Kumar (2015) used Random Forest Classifier to forecast diabetes behaviour. [4] Nawaz Mohamudally1 and Dost Muhammad (2011) used C4.5 decision tree algorithm, Neural Network, K-means clustering algorithm and Visualization to predict diabetes. [1]

Lee et al. [7] applied decision tree CART algorithm and emphasis on the class imbalance limitation before applying any algorithm to achieve better accuracy rates. K.VijiyaKumar et al. [12] proposed random Forest algorithm which gave the best results for diabetic prediction and showed that the prediction system can predict the diabetes disease effectively and efficiently. Nonso Nnamoko et al. [10] presented predicting diabetes onset: an ensemble supervised learning. They used 5 classifiers for ensembles and a meta-classifier to aggregate their outputs. Tejas N. Joshi et al. [13] proposes an effective technique for earlier detection of diabetes disease via 3 different supervised ML methods including: SVM, Logistic regression, ANN. Deeraj Shetty et al. [8] proposed a prediction solution using data mining assemble Intelligent Diabetes Disease Prediction System that gives analysis by utilizing diabetes patient's database.

They proposed Bayesian and KNN (K-Nearest Neighbor) algorithms to apply analyzing the patient's database and acquiring various attributes for diabetes prediction. Muhammad Azeem Sarwar et al. [11] proposed a research study on diabetes prediction which used 6 different algorithms. Comparison of these different techniques is utilized to reveal which algorithm is best suited for the prediction of diabetes.

3. Proposed Methodology

3.1. Dataset collection and description

The data is gathered from the Taipei Municipal medical center contains 15000 women aged between 20 and 80. Some suffer from diabetes and others are healthy. The dataset contains 8 attributes which are used to train the model. The ninth attribute is a class variable for each data point, it shows the outcome of the prediction where **diabetic:** 1 = diagnosed diabetes, 0 = not diagnosed diabetes.

No.	Attribute	Description
1	Pregnancies	pregnant frequency
2	Plasma Glucose	Plasma glucose concentration
3	Diastolic Blood Pressure	(mm Hg)
4	Triceps Thickness	(mm)
5	Serum Insulin	(mu U/ml)
6	BMI	(weight in kg/(height in m)^2)
7	Diabetes Pedigree	Probability of diabetes
8	Age	Age in years

Table 1. Dataset attributes - features.

3.2. Data analysis

Data analysis is an important process. To improve quality and effectiveness, the dataset was analyzed. Most of the columns are integers, except for BMI and Diabetes Pedigree are float numbers of type-based 64. The data do not contain missing values, which makes them ideal for training models. The data consists of 15000 entries and 10 columns. The mean age is approximately 30 years with BMI around 31.5. Plasma glucose values range from 44 to 192 with an average of 107, an outlier was detected in Serum insulin in 14 to 799.

After observation of the initial stages of analysis, the data set was imported in CSV format. Python library was utilized to handle and read the data efficiently. One of the primary libraries imported was Pandas which facilitated the loading of the CSV file into a structured Data-frame format, providing us with a tabular structure which will enable us to perform various preprocessing steps and analysis as shown below:

	PatientID	Pregnancies	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age	Diabetic
0	1354778	0	171	80	34	23	43.509726	1.213191	21	0
1	1147438	8	92	93	47	36	21.240576	0.158365	23	0
2	1640031	7	115	47	52	35	41.511523	0.079019	23	0
3	1883350	9	103	78	25	304	29.582192	1.282870	43	1
4	1424119	1	85	59	27	35	42.604536	0.549542	22	0
5	1619297	0	82	92	9	253	19.724160	0.103424	26	0
6	1660149	0	133	47	19	227	21.941357	0.174160	21	0
7	1458769	0	67	87	43	36	18.277723	0.236165	26	0
8	1201647	8	80	95	33	24	26.624929	0.443947	53	1
9	1403912	1	72	31	40	42	36.889576	0.103944	26	0
10	1943830	1	88	86	11	58	43.225041	0.230285	22	0
11	1824483	3	94	96	31	36	21.294479	0.259020	23	0
12	1848869	5	114	101	43	70	36.495320	0.079190	38	1
13	1669231	7	110	82	16	44	36.089293	0.281276	25	0
14	1683688	0	148	58	11	179	39.192076	0.160829	45	0

Figure 1. Diabetic Dataset.

Pandas library is crucial for organizing datasets, it grants access to wide range of functions for handling missing values, generating statistical analysis and preparing the data for visualizations.

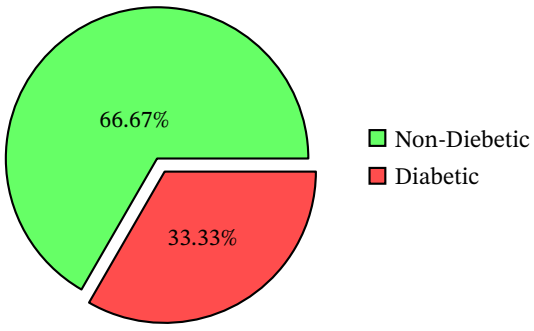


Figure 2. Dataset imbalance

The diabetic data shows a class imbalance, with about 33% of patients being diabetic and 67% are normal

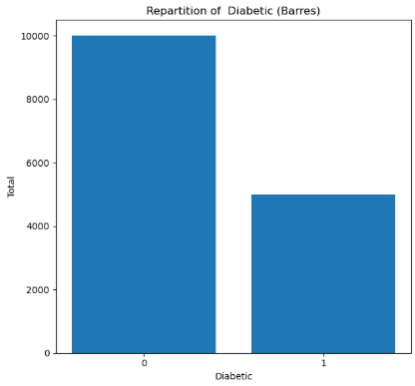


Figure 3. Ratio of Diabetic and Non Diabetic Patient.

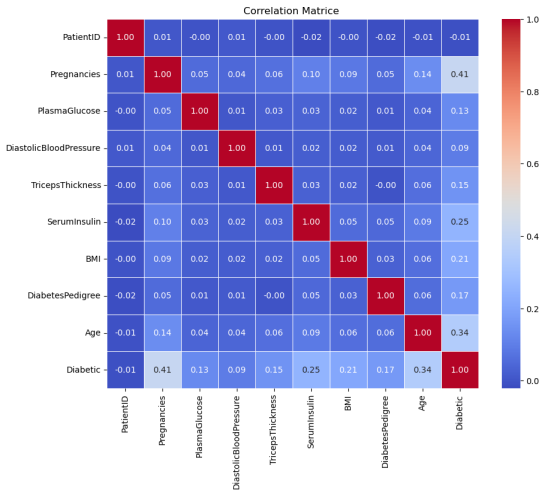


Figure 4. Correlation HeatMap.

The correlation matrix assists us in identifying which factors have the strongest relationships with the presence of diabetes. By examining the correlation matrix, Pregnancies show a strong correlation with diabetes (r=0.41, r=0.41), Serum Insulin with (r= 0.25, r= 0.25),), Age (r=0.34r = 0.34), whereas BMI (r=0.21r = 0.21) shows a moderate correlation. Overall, the heatmap matrix highlights factors such as pregnancies, age and insulin being more influential in predicting diabetes. Above are charts on the correlation on variables that has more influence on the target value

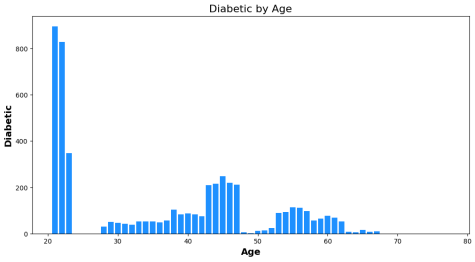


Figure 5. Diabetic by age.

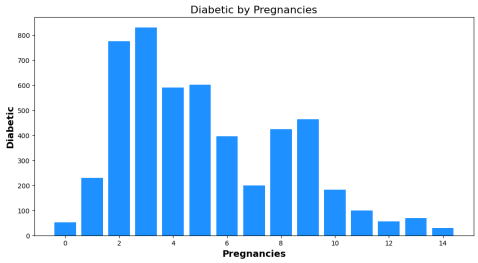


Figure 6. Diabetic by pregnancies.

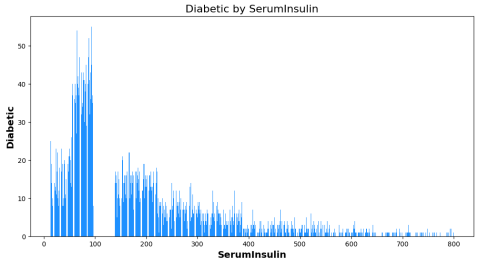


Figure 7. Diabetic by serum Insulin.

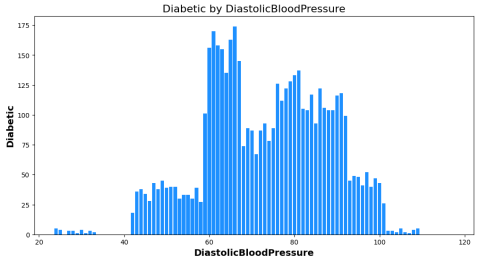


Figure 8. Diabetic by diastolic blood pressure.

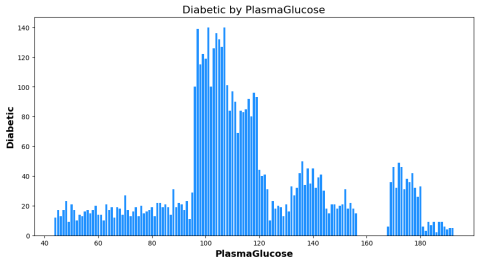


Figure 9. Diabetic plasma glucose.

The analysis shows that the "Patient ID" feature does not have any predictive value for our study, which will not benefit the insert in our training set; therefore, this feature will be excluded.

Furthermore, the observed correlations with the target variable "Diabetic" are least influential, with the exception of the "Pregnancies" feature, which revealed the strongest correlation. This low or non-existent correlation among other features suggests their independence, eliminating the need for a de-correlation step using dimensionality reduction techniques, such as Principal Component Analysis (PCA).

Based on these findings, the following hypotheses have been formulated:

- All variables, except for "Patient ID", contribute to explaining the target variable "Diabetic".
- The "Pregnancies" variable plays a significant role in predicting diabetes, as evidenced by its stronger correlation with the target.
- The "Plasma Glucose", "BMI", and "Age" variables are closely linked to known metabolic factors and may have an indirect influence on diabetes.
- The "Patient ID" and "Diastolic Blood Pressure" variables provide no significant information for prediction and can be excluded.

In validating these hypotheses, statistical tests will be conducted to select the relevant features, with the development of multiple models to identifying the best predictive model.

4. Feature Engineering and Selection

4.1. Data preprocessing

Data preprocessing is a crucial step in building robust and accurate models. observation, the dataset contains no missing values which eliminates the need for imputation strategies and ensuring data quality. To enhance model's performance, feature engineering was applied to identify the most influential features that contributes to the prediction of diabetes. for this reason, both linear and polynomial multiple regression were applied, enabling comprehensive evaluation of the relationships between the features and the target variable. Given the first result for multiple regression for linear:

OLS Regression Results

Dep. Variable:	Diabetic	R-squared:	0.345
Model:	OLS	Adj. R-squared:	0.345
Method:	Least Squares	F-statistic:	879.0
Date:	Fri, 21 Mar 2025	Prob (F-statistic):	0.00
Time:	00:15:07	Log-Likelihood:	-6825.0
No. Observations:	15000	AIC:	1.367e+04
Df Residuals:	14990	BIC:	1.375e+04
Df Model:	9		
Covariance Type:	nonrobust		

Figure 10. Multiple regression for linear.

The F-statistic of 879 with p-value of 0 suggests that at least one feature has a meaningful relationship with diabetes. Among the features, **Pregnancies** has the strongest positive relationship with diabetes, followed by **Plasma Glucose**, **DiastolicBloodPressure**, **Serum Insulin**, **Age** and **BMI**, all which are correlated with diabetes. The **Patient ID** shows no significance so we will remove this column from the training set. These findings suggest that these characteristics play a crucial role in predicting diabetes.

OLS Regression Results

Dep. Variable:	Diabetic	R-squared:	0.428
Model:	OLS	Adj. R-squared:	0.428
Method:	Least Squares	F-statistic:	623.8
Date:	Fri, 21 Mar 2025	Prob (F-statistic):	0.00
Time:	00:31:57	Log-Likelihood:	-5808.3
No. Observations:	15000	AIC:	1.165e+04
Df Residuals:	14981	BIC:	1.180e+04
Df Model:	18		
Covariance Type:	nonrobust		

Figure 11. Multiple regression for polynomial.

Given that real-world data produce non-linear patterns, introducing polynomial regression incorporates higher-order terms into the dataset, enhancing the model's ability to model non-linear relationships which improves the model's accuracy and predictive ability. The results from the polynomial multiple regression model indicate an improvement in the model's performance. The **R-square shows a value of 0.428** which is higher than the previous linear model, this conforms that the model stays remains robust even after adjusting the number of features, preventing overfitting.

The decrease in the **log-likelihood value of -5808.3** compared to linear model reflects a more complex model with more parameters, which aligns with the inclusion of polynomial terms for capturing non-linear relationships. These findings support the hypothesis that incorporation of polynomial terms improves the model's capacity to capture patterns in data, leading to an improved predictive performance.

4.2. Verification of variance inflation factor

The variance inflation factor **VIF** is used to detect multicollinearity in regression models; this usually occurs when 2 or more features (independent variables) are strongly correlated. Multicollinearity can sometimes distort the estimated coefficients, leading to difficulty to determine the true influence of each variable.

$$VIF_i = \frac{1}{1 - R_i^2} \tag{1}$$

Where:

- VIF_i is the Variance Inflation Factor for the i-th predictor.
- R_i^2 is the coefficient of determination from regressing the i-th predictor on all other predictors.

Result:

	Feature	VIF
0	const	47.373366
1	Pregnancies	1.041960
8	Age	1.033993
5	SerumInsulin	1.020842
6	BMI	1.012699
4	TricepsThickness	1.008115
7	DiabetesPedigree	1.007335
2	PlasmaGlucose	1.005320
3	DiastolicBloodPressure	1.003652

Figure 12. Variance Inflation Factor

For all features except the constant have values close to 1, meaning there's no multicollinearity between the features, indicating that each feature has a unique information for the model. The constant has a high VIF of 47.37 but it wont affect the predictor's relationship with the model. All features can be kept and the next step is exploring non-linear patterns to improve the model.

4.3. Splitting the data

The dataset is divided into 3 parts: **training, validation, and testing.**. Initially the data is split into training set which consist of 70% and a temporary set of 30% using the "train test split" from scikit learn library. The training set is used to train the model, while the temporary is divided to create a validation set and test set. The temporary set is split equally between validation and testing. This ensures that 70% of the data is utilized for training, 15% for validation and 15% for testing. The validation set is important to fine-tune the model and prevent future overfitting, while the testing set is used to evaluates the model's efficiency on unseen data.

Table 2. Division of data

Data	Split Division	Shape
Training set	70%	(10500, 8)
Validation set	15%	(2250, 8)
Testing set	15%	(2250, 8)

Note: The table contains information on dividing data.

5. Applying Machine Learning Models

5.1. Linear Regression

The linear regression model is a statistical method utilized to find the relationship between dependent variable (target value) and several independent variable (features). It does that by fitting a linear equation to the data to find the best fitting line which minimizes the difference between the actual values and predicted values. The formula is a simple linear regression with one feature variable:

$$y = \beta_0 + \beta_1 x + \epsilon \tag{2}$$

Where:

- y = Target variable - dependent variable
- β_0 = Intercept (value of y when $x = 0$)
- β_1 = slope, representing the change in y for a one-unit change in x
- x = Feature variable - independent variable
- ϵ = Error term (captures the noise or unexplained variability in the model)

5.2. Logistic Regression

The logistic regression is a machine learning algorithm mainly used for classification tasks where the target value is often binary or categorical. In our case here, the target value (diabetic or non-diabetic). Unlike the linear regression model which finds the relationship between the independent and dependent variables, the logistic regression uses a transformation process which applies the sigmoid function to ensure the output values range between 0 and 1. The logistic regression model is expressed as:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n)}} \tag{3}$$

Where:

- $P(Y = 1|X)$ The probability that the target variable Y equals 1 given the input X.
- β_0 Intercept.
- $\beta_1, \beta_2, ..., \beta_n$ The coefficients.
- $X_1, X_2, ..., X_n$ Input features.
- e Euler's number | 2.718.

The sigmoid function which maps a real number to a value between 0 and 1, is given by:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

Where:

$$z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n \tag{5}$$

5.3. Decision Tree model

The Decision Tree is known as a supervised machine learning algorithm mainly utilized for both classification and regression tasks. It works by recursively splitting the dataset into subsets based on the input features value, resulting in a tree-like structure. Each node represents a decision based on a feature and each branch corresponds to the outcome of that decision and each leaf node represents the final output of the prediction.

The Decision Tree model can be utilized to classify the patients as diabetic or non-diabetic by learning the decision rules given from the features such as **Plasma Glucose**, **BMI** and **Pregnancies**. The splitting criterion of each node is determined by using different impurity measure such as MSE (Mean Squared Error), Entropy and Gini Index.

Gini Indicis calculated as:

$$Gini = 1 - \sum_{i=1}^{n} p_i^2 \tag{6}$$

Where p_i is the probability of class i at a given node.

Entropy measures the uncertainty:

$$Entropy = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{7}$$

Information Gain measures the reduction in entropy after a dataset is split on a particular feature:

$$IG = Entropy(parent) - \sum \frac{N_k}{N} Entropy(k) \tag{8}$$

Where N_k is the number of samples in the k-th subset, and N is the total number of samples.

5.4. Support Vector Machine | SVM

The Support Vector Machine is a very powerful supervised learning algorithm utilized for classification and regression tasks. In the context of diabetes prediction, SVMs are efficient because they excel at finding optimal boundaries between diabetic and non-diabetic based on multiple feature variables.

SVMs functions by finding the best hyperplane which separates the data points into distinct classes.This hyperplane is a simple in a 2D-dimensional space, it becomes a hyperplane in higher dimensions and the goal is to maximize the distance between the hyperplane and the closest data points is called support vectors. After training, the model learns from past patient data where each patient is represented by the given features.

The equation of the hyperplane is given by:

$$w \cdot x + b = 0 \tag{9}$$

Where:

- w = Weight vector
- x = Input vector
- b = Bias

The decision function is:

$$f(x) = \text{sign}(w \cdot x + b) \tag{10}$$

If $f(x) > 0$, the patient is classified as diabetic; otherwise, non-diabetic.

The objective function becomes:

$$\min\left(\frac{1}{2}\|w\|2 + C\sum \xi_i\right) \tag{11}$$

Where:

- C = Regularization parameter

In this study, different values of C were tested: 1, 10, and 100. The impact of these values is as follows:

- $C = 1$
- $C = 10$
- $C = 100$

By comparing these models, we can determine which value of C gives the best performance.

5.5. Neural Networks

A Neural Network with 3 layers is a type of artificial neural network (ANN) which consists of input layers, one hidden layer and an output layer. This simple structure makes it a simple feedforward neural network, where the information flows from input to output direction without any loops or cycles.In the context of diabetes prediction, a neural network with 3 simple layers is constructed for diabetes prediction. The process includes data normalization and sequential creation. The first step involves normalizing the data using Standard-scaler from **sklearn.preprocessing** function. This technique ensures all features have a mean of 0 and a standard deviation of 1, making the training process stable and prevents the model from being biased.

The training set was used to fit the scaler, the validation set and testing set were transformed using the same scaler to avoid data leakage. The neural network model was defined using the Sequential API from tensorflow.Keras. It consists of:

- Input Layer

$$Z_1 = W_1 \cdot X + b_1$$

$$A_1 = ReLU(Z_1) = \max(0, Z_1)$$

- Hidden Layer

$$Z_2 = W_2 \cdot A_1 + b_2$$

$$A_2 = ReLU(Z_2)$$

- Output Layer

$$Z_3 = W_3 \cdot A_2 + b_3$$

$$\hat{y} = Softmax(Z_3) = \frac{e^{Z_3}}{\sum e^{Z_3}}$$

The **ReLU activation** helps the neural network learn non-linear relationships, while **Softmax activation** facilitates probabilistic predictions for the 3 classes.

5.6. K-Nearest Neighbors

K-Nearest classifier is simple machine learning algorithm mainly for classification tasks. It works on finding the K data points to a given a data point and then assigning the common class among those neighbors. It computes the distance between a test point and all the training points and selecting the K nearest neighbors. The most common metrics utilized are Euclidean, Manhattan and Malinowski distance.

In the context of diabetes prediction, The KNN can be used to classify the patients into categories like "diabetic" or "non-diabetic" based of various input features.

Euclidean Distance: For two points $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$, the Euclidean distance $d(p, q)$ is given by:

$$d(p, q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

Where:

- p_i and q_i are the features of the two points,
- n is the number of features.

5.7. Gradient Boosting Classifier (GCB)

The Gradient Boosting Classifier is a an ensemble machine learning technique that build a strong predictive model by combining weaker models in a sequential manner to correct errors from previous iterations. It minimizes the loss function by applying gradient descent, making the model highly effective to handle complex datasets.

The model can be trained on medical features such as: Age, BMI, blood pressure, glucose levels, insulin, etc.

Loss Function and Gradient Descent: The goal minimizing the loss function $L(y, F(x))$, where y is the actual value and $F(x)$ is the predicted value. For regression,**Mean Squared Error (MSE)**:

$$L(y, F(x)) = (y - F(x))^2$$

For binary classification, **Binary Cross-Entropy**:

$$L(y, F(x)) = -y \log(F(x)) - (1 - y) \log(1 - F(x))$$

where y is the actual class label (0 or 1) and $F(x)$ is the predicted probability.

The Loss Function gradient: For model improvement, the gradient of the loss function is computed with respect to the model's predictions:

$$g_i = \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}$$

For binary cross-entropy, the gradient is:

$$g_i = F(x_i) - y_i$$

where g_i is the gradient for the i-th sample.

A decision tree is then trained on the negative gradients g_i. Denoted as $h_t(x)$:

$$h_t(x) = -g_i$$

The models prediction are updated by adding the output of the new tree scaled by a learning rate η:

$$F_t(x) = F_{t-1}(x) + \eta \cdot h_t(x)$$

where:

- $F_{t-1}(x)$ is the previous prediction,
- $h_t(x)$ is the prediction of the new tree,
- η is the learning rate that controls the step size.

Final Model Prediction

After T boosting steps, the final model prediction is given by the sum of all the trees' predictions:

$$F_T(x) = F_0(x) + \sum_{t=1}^{T} \eta \cdot h_t(x)$$

where $F_0(x)$ is the initial model prediction (often the mean of the target values), and $h_t(x)$ is the output of the t-th tree.

5.8. Overview of Random Forest

Random Forest is an ensemble method that uses multiple decision trees to make a prediction. Each tree is trained on a random subset of the data and the final prediction is based on the predictions from all trees.

Formulas for classification tasks:

Gini (Classification): Gini Impurity helps decide how to split the data at each node by measuring of how mixed the classes are at a node. The formula is:

$$Gini(t) = 1 - \sum_{k=1}^{K} p_k^2$$

Where:

- p_k is the probability of class k in the node.
- K is the total number of classes.

Entropy (Classification): Entropy is another measure for deciding the best split, which measures the disorder or uncertainty of the data:

$$Entropy(t) = -\sum_{k=1}^{K} p_k \log_2(p_k)$$

Where:

- p_k is the probability of class k.

Final prediction is made by taking a majority vote from all the trees:

$$\hat{y} = \text{Majority Vote}(y_1, y_2, \ldots, y_T)$$

Where:

- y_i is the predicted class from the i-th tree.
- T is the total number of trees.

6. Evaluation Metrics

we use several evaluation metrics to measure how well our model's performance, below are the used metrics:

6.1. Precision

which is the proportion of positive predictions that are actually correct.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Where:

- TP = True Positives (correctly predicted positive cases)
- FP = False Positives (incorrectly predicted as positive)

6.2. Recall

Recall is the proportion of actual positive cases that were correctly identified by the model.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where:

- TP = True Positives (correctly predicted positive cases)
- FN = False Negatives (incorrectly predicted as negative)

6.3. F1-Score

The F1-Score is the harmonic mean of precision and recall, it balances the two metrics and provides a single score to evaluate the model's performance.

$$F1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-Score ranges from 0 to 1, with 1 being the best possible performance.

6.4. Support

Support represents how many times a particular class appears in the data.

$$\text{Support} = \text{Count of true instances for each class}$$

6.5. Confusion Matrix

A confusion matrix is a table that is used to evaluate the performance of a classification model. It compares the predicted labels with the true labels.

For binary classification, the confusion matrix looks like this:

	Predicted Positive	Predicted Negative
True Positive	TP	FN
True Negative	FP	TN

Where:

- TP = True Positives (correctly predicted positive cases)
- TN = True Negatives (correctly predicted negative cases)
- FP = False Positives (incorrectly predicted as positive)
- FN = False Negatives (incorrectly predicted as negative)

7. Experimental Results

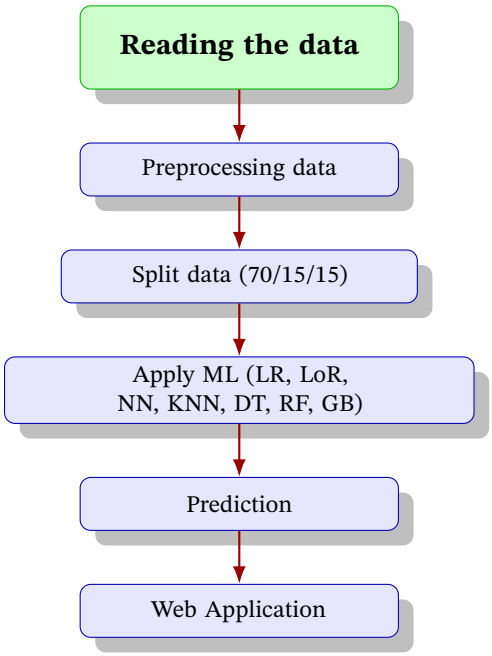
7.1. Model Building Phase for Diabetes Prediction

This phase includes building several models for prediction of diabetes. In this phase, we have implemented various machine learning algorithms which are discussed above, below are the steps

- **Step 1:** Libraries have been imported, and the dataset has been loaded.
- **Step 2:** The data has been preprocessed and normalized to ensure it's suitable for model training.
- **Step 3:** A percentage split has been performed: 70% for the training set, 15% for the validation set, and 15% for the testing set.
- **Step 4:** The following machine learning algorithms have been imported: K Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree, Logistic Regression, Random Forest, and Gradient Boosting Algorithm.
- **Step 5:** Models are trained using the training set and tested using the testing set.
- **Step 6:** A comparison and evaluation of the models are performed to determine which model performed the best.

7.2. Experiment workflow

As shown below, is the workflow taken to conduct experimental testing on the models:



8. Result Metrics

8.1. Neural Network

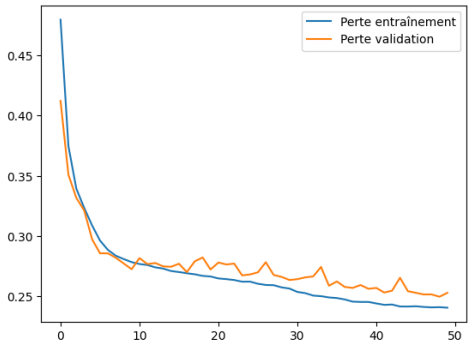


Figure 13. Validation metric over epochs.

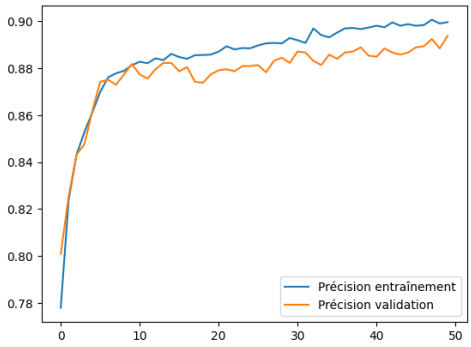


Figure 14. Test metric over epochs.

8.2. Support Vector Machine

8.3. Accuracy metrics

Below is a table that sumerizes the accuracy metrics for each model in details. We can see that Gradient Boosting algorithm has the highest accuracy metric among the other models and the logistic regression has the lowest one.

Table 3. Model Metrics

Model	Set	Accuracy	MSE / R^2
LR	Val	-	MSE: 0.1489, R^2: 0.3412
	Test	-	MSE: 0.1428, R^2: 0.3412
LoR	Val	0.78	-
	Test	0.7916	-
DT	Val	0.8978	-
	Test	0.8880	-
NN	Test	0.89	-
KNN	Val	0.8347	-
	Test	0.8391	-
GBC	Val	0.9524	-
	Test	0.9427	-
RF	Val	0.9329	-
	Test	0.9338	-

8.4. Confusion matrix

As we can see above are the accuracy metrics and the confusion matrix for each model, this will help us later for determining which

Table 4. Confusion Matrices

Model	Set	TP	FP	FN	TN
LoR	Val	1320	153	342	435
	Test	1365	167	302	416
DT	Val	1345	128	102	675
	Test	1371	161	91	627
KNN	Val	1316	157	215	562
	Test	1371	161	201	517
GBC	Val	1429	44	63	714
	Test	1474	58	71	647

model performs the best among the others. For this reason, we can conclude that a one model performs better than the other, later in the conclusion section we will discuss which model performed the best and why.

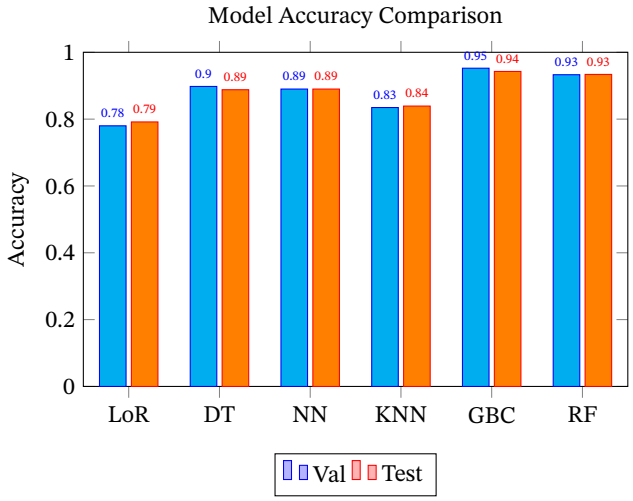


Figure 15. Validation and Testing Accuracy of Each Model

9. Building an Application

For the development of web applications, we developed a robust application that integrates the trained prediction model. We used Python as our main programming language and to ensure both functionality and efficiency, e utilized a variety of dependencies.

The backend was built using **FastAPI (0.115.11)** and **Flask (3.1.0)** which provide a strong framework to handle API requests and serve predictions. Model loading was managed with **joblib (1.4.2)** while **TensorFlow (2.19.0)**, **Keras (3.6.0)** developed the core of the machine learning pipeline. Data preprocessing was made using **Pandas (2.2.3)**, **Numpy (1.26.4)** with **Scikit-Learn (1.6.1)** that supported the training and evaluation of the model. For data visualization, **Matplotlib (3.10.1)** and **Seaborn (0.13.2)** were used to create the charts with the incorporation of **Stats-models (0.14.4)** for statistical analysis.

To establish the API communication, we used Requests (2.32.3) whereas Uvicorn (0.34.0) served as an ASGI server, all these dependencies ensure the deployment of the application. After combining all these technologies, we were able to build a user-friendly application for the prediction of diabetes.

10. Conclusion

From this study, we can conclude that after experimenting with various machine learning models for diabetes prediction, Linear regression model explained 34% of the data with an average of 14% error, leaving 60% of the other cases unexplained. The Logistic regression a good metric of Recall, precision and F1-score of 78% leaving a 32% of other cases unexplained, making it unsuitable to set it as a good predictive model. The Decision Tree model achieved a better Recall, precision and F1-score of 90% but still was not optimal. The Support Vector Machine with an RBF kernel and c=100 had a better accuracy success rate of 83% and 90% without overfitting or underfitting.

The KNN model achieved 83% of precision, showing a good generalizability but also struggled with class 1 maybe due to imbalance. Finally, Gradient Boosting outperformed all the other models with more 90% accuracy, high F1 scores for both classes, and a consistent positive result for the validation set and testing set achieving a 95. 24% accuracy. Therefore, Gradient Boosting is considered the best performing model for diabetes prediction offering the hightest efficiency and generalizability.

References

[1] D. M. Khan and N. Mohamudally, "An integration of k-means and decision tree (id3) towards a more efficient data mining algorithm", *Journal of Computing*, vol. 3, no. 12, pp. 76–82, 2011.

[2] K. Rajesh and V. Sangeetha, "Application of data mining methods and techniques for diabetes diagnosis", *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 3, pp. 224–229, 2012.

[3] A. A. Aljumah, M. G. Ahamad, and M. K. Siddiqui, "Application of data mining: Diabetes health care in young and old patients", *Journal of King Saud University-Computer and Information Sciences*, vol. 25, no. 2, pp. 127–136, 2013.

[4] M. Butwall and S. Kumar, "A data mining approach for the diagnosis of diabetes mellitus using random forest classifier", *International Journal of Computer Applications*, vol. 120, no. 8, 2015.

[5] T. Eswari, P. Sampath, S. Lavanya, *et al.*, "Predictive methodology for diabetic data analysis in big data", *Procedia Computer Science*, vol. 50, pp. 203–208, 2015.

[6] A. Iyer, S. Jeyalatha, and R. Sumbaly, "Diagnosis of diabetes using classification mining techniques", *arXiv preprint arXiv:1502.03774*, 2015.

[7] D. K. Choubey, S. Paul, S. Kumar, and S. Kumar, "Classification of pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection", in *Communication and computing systems: proceedings of the international conference on communication and computing system (ICCCS 2016)*, 2017, pp. 451–455.

[8] D. Shetty, K. Rit, S. Shaikh, and N. Patil, "Diabetes disease prediction using data mining", in *2017 international conference on innovations in information, embedded and communication systems (ICIIECS)*, IEEE, 2017, pp. 1–5.

[9] D. Dutta, D. Paul, and P. Ghosh, "Analysing feature importances for diabetes prediction using machine learning", *IEEE*, pp. 924–928, 2018. DOI: 10.1109/IEMCON.2018.8614871.

[10] N. Nnamoko, A. Hussain, and D. England, "Predicting diabetes onset: An ensemble supervised learning approach", in *2018 IEEE Congress on evolutionary computation (CEC)*, IEEE, 2018, pp. 1–7.

[11] M. A. Sarwar, N. Kamal, W. Hamid, and M. A. Shah, "Prediction of diabetes using machine learning algorithms in healthcare", in *2018 24th international conference on automation and computing (ICAC)*, IEEE, 2018, pp. 1–6.

[12] K. VijiyaKumar, B. Lavanya, I. Nirmala, and S. S. Caroline, "Random forest algorithm for the prediction of diabetes", in *2019 IEEE international conference on system, computation, automation and networking (ICSCAN)*, IEEE, 2019, pp. 1–5.

[13] S. R. Joshi, *RSSDI Textbook of Diabetes Mellitus*. Jaypee Brothers Medical Publishers, 2020.